

## Cloud API: TechSolver verslag

### Facebook API

Wanneer ik op login pagina op de facebook login knop druk. Dan wordt de 'FBLogin()' functie aangeroepen en met het stukje code hieronder wordt eerst de facebook sdk opgehaald.

```
(function(d, s, id){  
  var js, fjs = d.getElementsByTagName(s)[0];  
  if (d.getElementById(id)) {return;}  
  js = d.createElement(s); js.id = id;  
  js.src = "//connect.facebook.net/en_US/sdk.js#xfbml=1&version=v2.5&appId=1526991634270394";  
  fjs.parentNode.insertBefore(js, fjs);  
}(document, 'script', 'facebook-jssdk'));
```

Dan gaat hij de API aanroepen om zo de gegevens van facebook van zijn account te halen. Voor hij effectief de gegevens krijgt, gebruikte hij deze URI bij mijn account met een bepaald token. Dit gebeurt via de GET methode om zo een token te verkrijgen en zo een sessie te starten met het ingelogde account.

Request URL: [https://graph.facebook.com/v2.5/me?access\\_token=EAAVsynHviLoBAGpUZAiVMh16fMZBcb3j9Yq4ficUzjdrGkrVZBQ8brVInFyiZBZAZAcIPzVv4692o9DZAgFpKsgSyyvmNIoG3DM7kkASlh0uSPLeQNVJizcW7oI1Yb02Ehg6gZBZB4s3HPpfb4x16C5RST2ivAC80Virs5SP254C7QZDZD&callback=FB.\\_\\_globalCallbacks.f177a8bd70eaa28&method=get&pretty=0&sdk=joey](https://graph.facebook.com/v2.5/me?access_token=EAAVsynHviLoBAGpUZAiVMh16fMZBcb3j9Yq4ficUzjdrGkrVZBQ8brVInFyiZBZAZAcIPzVv4692o9DZAgFpKsgSyyvmNIoG3DM7kkASlh0uSPLeQNVJizcW7oI1Yb02Ehg6gZBZB4s3HPpfb4x16C5RST2ivAC80Virs5SP254C7QZDZD&callback=FB.__globalCallbacks.f177a8bd70eaa28&method=get&pretty=0&sdk=joey)

Request Method: GET

Status Code: 200

### REST API

Eerst en vooral dit heb ik verkeerd begrepen en dus ook verkeerd aangepakt. In plaats van dat ik zelf API calls moest maken met eigen API op onze webapplicatie via GET, POST,.. heb ik API calls gemaakt van en naar firebase.

In de onderliggende code doe ik een POST naar een bepaalde locatie in firebase (in een map test met de geuploadede file die test.json zal genoemd worden)

Je kan een file van je pc naar firebase uploaden op die manier.

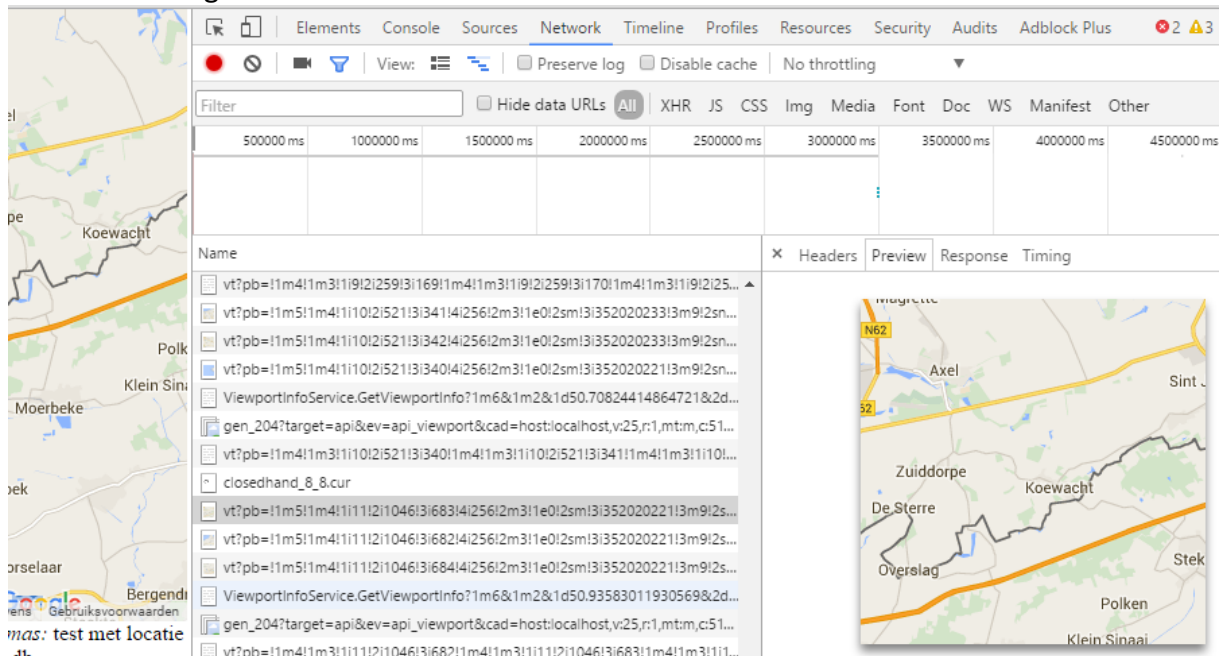
```
<form method="post" enctype="multipart/form-data" action="https://firebasestorage.googleapis.com/v0/b/resplendent-inferno-9134.appspot.com/o/test%2Ftest.json">  
  <input type="file" name="chooseFile" id="fileChosen" value="Choose file">  
  <input type="submit" name="uploadFile" value="POST">  
</form>
```

In de onderliggende code zullen we een GET aanroepen naar een locatie van een file op firebase en deze opslagen op onze eigen files van onze pc.

```
<a href="https://firebasestorage.googleapis.com/v0/b/resplendent-inferno-9134.appspot.com/o/test%2Fabout.json?alt=media&token=746af4af-91ce-4403-9dac-a2ab53e19732" download="test">GET</a>
```

#### De VT?PB="" call:

Met deze call wordt de map "ineen gepuzzeld". Dit is de vaakst gebruikte call van de maps api. De call begint altijd met vt?pb. Als men uitzoomed wordt deze call dus ook meerdermaals gemaakt.



#### De CSI call:

Met deze call roept de maps API al de nodige statische documenten aan, javascript, css,... Google hoopt hiermee de nodige bandbreedte te verkleinen en snelheid te verhogen. ik denk dat dit komt doordat de webbrowser nu meerdere connecties kan gebruiken.

#### De Jp?id call:

hiermee word de api "bestuurd" dankzei deze calls weet de map hoever deze uitgezoomed moet zijn, waar er markers open moeten staan en wat de Z-index van deze markers is, routebegeleidingen enzovoort.

#### De authenticationService-call:

Met deze call wordt gechecked of er een API-key aanwezig is in de applicatie indien dit niet zo is geeft de console een google maps API warning weer.

#### Firebase api:

#### REST-server:

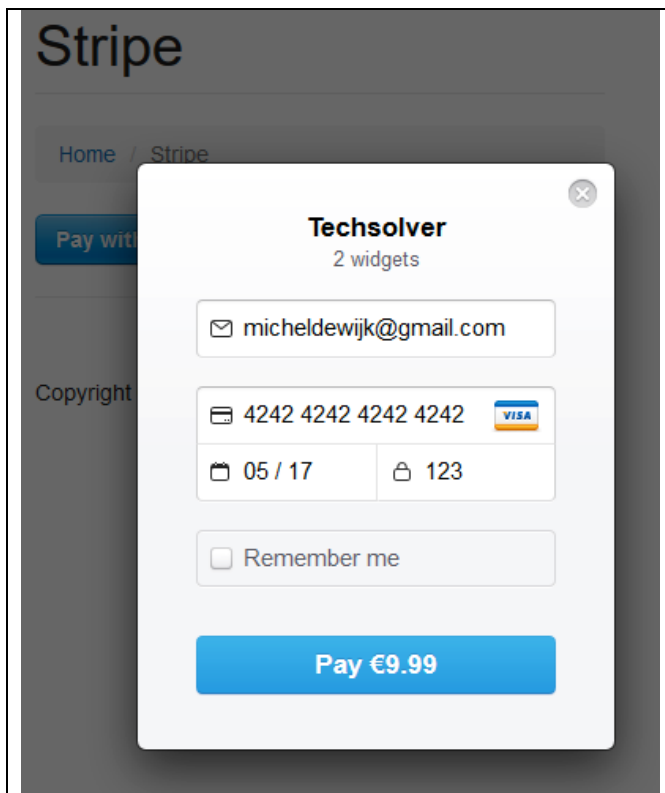
De frontend communiceert rechtstreeks met de backend via een REST server. Hierdoor wordt er geen informatie weergegeven onder de netwerktab.

stripe.html

```
<!-- <form action="javascript:payment()" method="POST"> -->
<form action="charge.php" method="POST">
  <script
    src="https://checkout.stripe.com/checkout.js" class="stripe-button"
    data-key="pk_test_QJp8r5q6HGjvCcsHrzprP6Xp"
    data-amount="999"
    data-name="Techsolver"
    data-description="2 widgets"
    data-image="/img/documentation/checkout/marketplace.png"
    data-locale="auto"
    data-currency="eur">
  </script>
</form>
```

De embedded Stripe form in **stripe.html** wordt aangeroepen via de website van Stripe met https, deze geeft een form terug met afhankelijk van de ingestelde waardes. Zoals bijvoorbeeld de **data-amount** bepaald hoeveel de form weergeeft om te betalen en de **data-currency** welke valuta getoont wordt. Door op **Pay** te drukken stuurt de form een **POST** naar **charge.php**.

stripe.html



```
<?php
    $token = $_POST['stripeToken'];
    echo ($token);
?>
```

De bedoeling was dit in plaats daarvan naar de javascript methode **payment()** ging in **server.js**.

Maar om te testen of de **stripe.html** form werkelijk werkte en een **POST** stuurde haalt deze de waarde van **stripeToken** op van de website van Stripe die deze genereert als de **Pay** succesvol is.

Deze **\$token** kan eenmalig gebruikt worden door de **server.js**.

charge.php

```
tok_18HIP4CoSFr0j34WARI2dCs6
```

server.js

```
function payment()
{
    alert("payment")
    //var stripeToken = request.body.stripeToken;
    var stripeToken = "tok_18HIP4CoSFr0j34WARI2dCs6";

    var charge = stripe.charges.create({
        amount: 999, // amount in cents, again
        currency: "eur",
        source: stripeToken,
        description: "Example charge"
    }, function(err, charge) {
        if (err && err.type === 'StripeCardError') {
            // The card has been declined
        }
    });
}
```

De **server.js** functie **payment()** verwerkt de **stripeToken** en maakt een **charge** doormiddel deze naar Stripe te sturen met het **currency** en de **stripeToken**. Deze functie werkt enkel via handmatig de generated **charge.php** token in de **payment()** functie plaatsen en de **server.js** te starten met **node js**.

De **stripe.html** hoort de **stripeToken** direct naar de **server.js** sturen maar dit is niet gelukt om toe te passen.

server.js

```
var server = http.createServer(function(req, res)
{
    res.writeHead(200, {
        'Content-Type': 'text/html'
    });

    res.write('<!doctype html>\n<html lang="en">\n' +
        '\n<meta charset="utf-8">\n<title>Test web page on node.js</title>\n' +
        '<style type="text/css">* {font-family:arial, sans-serif;}</style>\n' +
        '\n\n<h1>NodeJS Server Page</h1>\n' +
        '<div id="content"><p>Content:</p><ul><li>None</li></ul></div>' +
        '\n\n');
    res.end();

    payment();

}).listen(3000, '127.0.0.1');
console.log('Server running at http://127.0.0.1:3000');
```